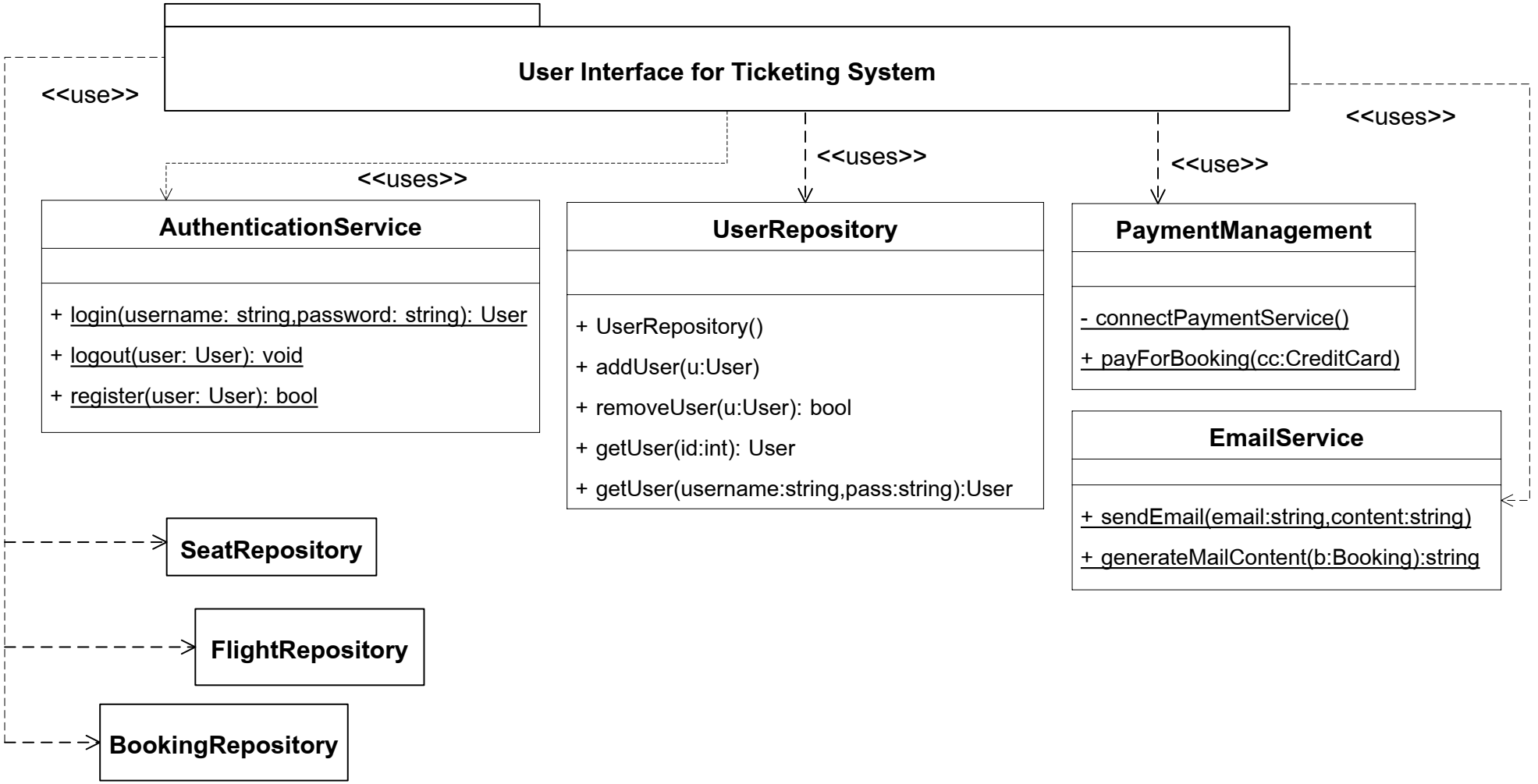
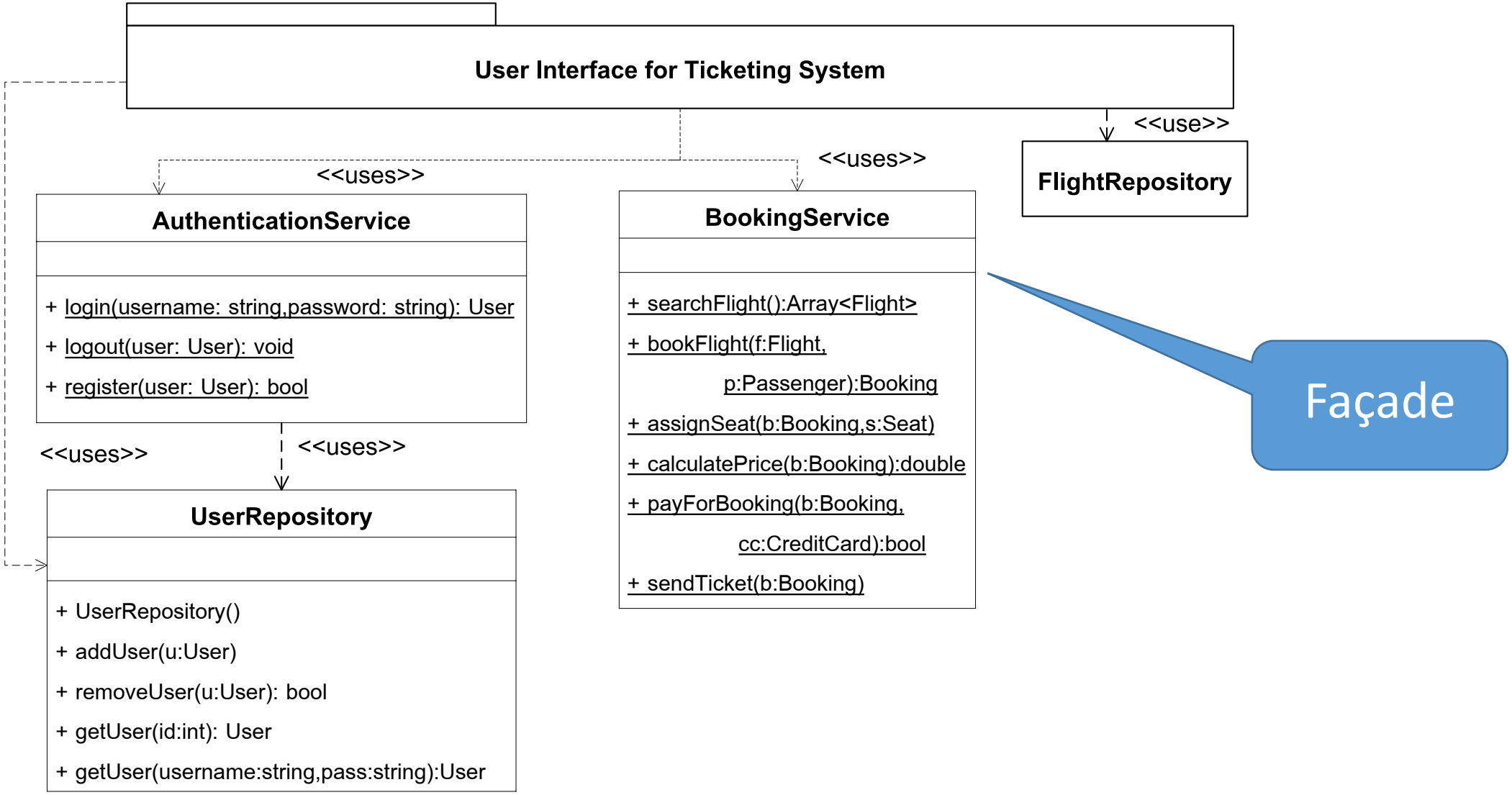
เดิม (ระหว่าง UI กับ BL)

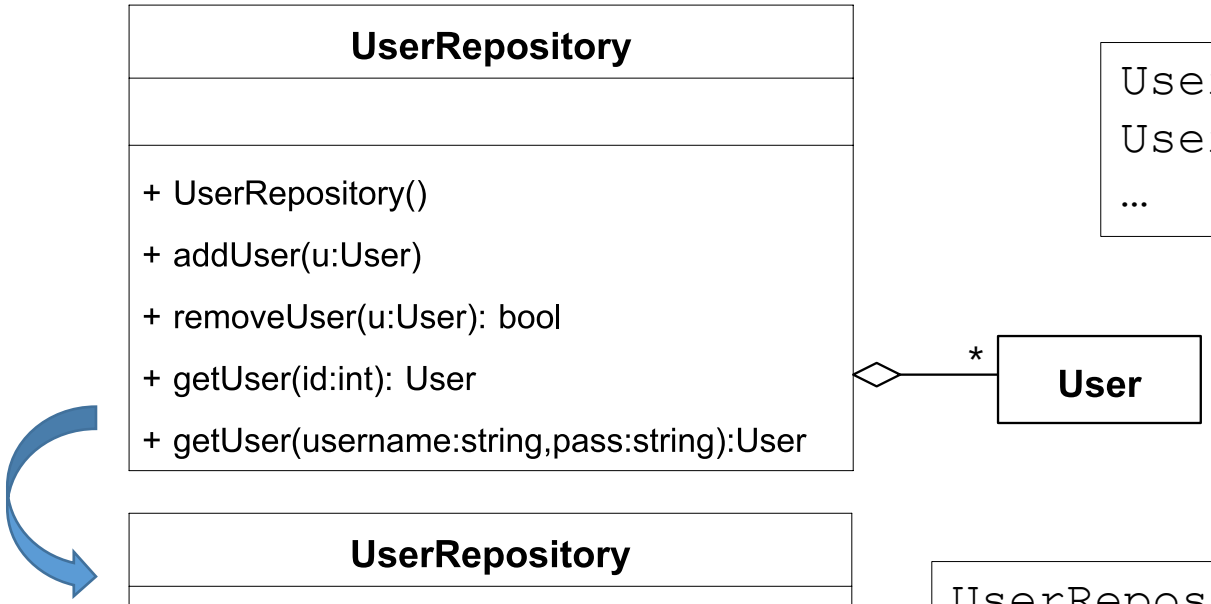


1. สร้าง Façade สำหรับการจองให้ UI ใช้ง่ายขึ้น



2. Repository **ทุกตัว** ทำให้เป็น Singleton ได้

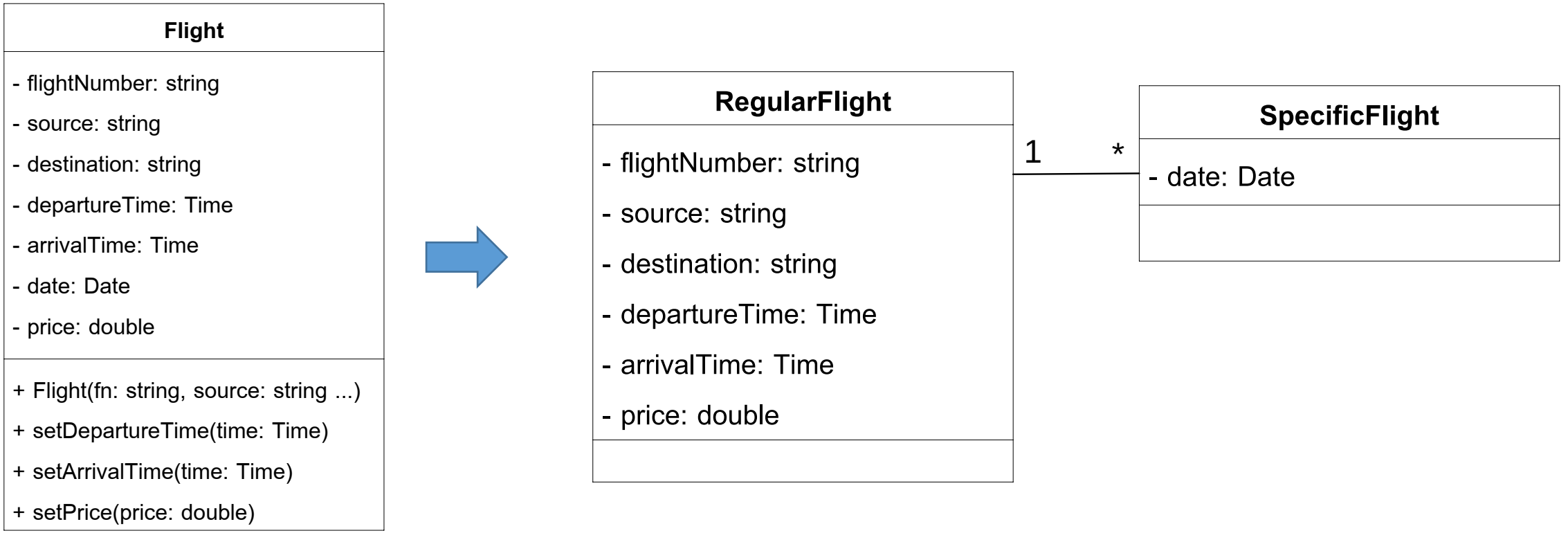
อาจสร้างและทำลาย
ในทุกครั้งที่ใช้



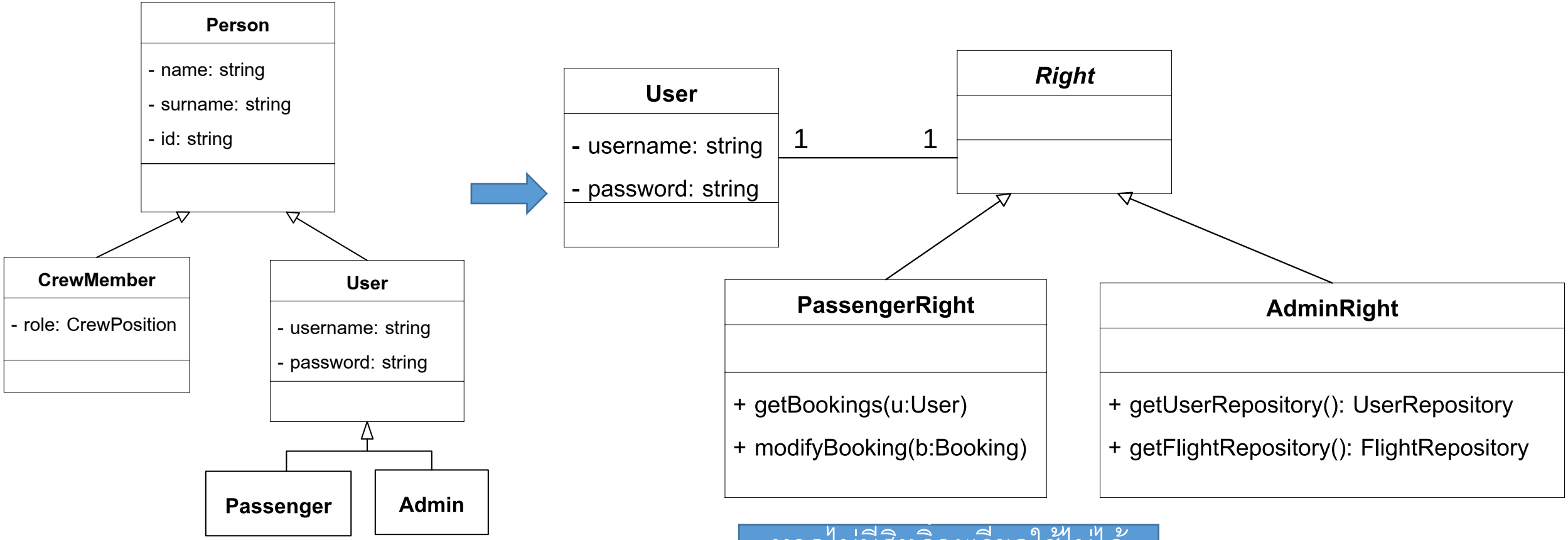

```
UserRepository ur = new UserRepository();
User admin = ur.getUser("admin", "admin");
…
```

```
UserRepository ur = UserRepository.getInstance();
User admin = ur.getUser("admin", "admin");
…
```

3. แบ่ง Flight ออกเป็น 2 คลาส ด้วย Abstraction-Occurence



4. Player-Role สำหรับ User แบบต่างๆ




```
if (user instanceof Admin)
{  UserRepository.getInstance().deleteUser(u1);
   …
}
```

หากไม่มีสิทธิจะเรียกใช้ไม่ได้

```
AdminRight r = (AdminRight) user.getRight();
r.getUserRepository().deleteUser(u1);
```

ส่งผลให้ UI เรียกใช้ Repository ต่างๆ โดยอ้อม แต่เรียกผ่าน User แทน (จาก AuthenticationService)